



(12) 发明专利申请

(10) 申请公布号 CN 111971663 A

(43) 申请公布日 2020.11.20

(21) 申请号 201880092394.4

(22) 申请日 2018.06.25

(30) 优先权数据

2018/02467 2018.04.16 ZA

(85) PCT国际申请进入国家阶段日

2020.10.14

(86) PCT国际申请的申请数据

PCT/IB2018/054660 2018.06.25

(87) PCT国际申请的公布数据

WO2019/202371 EN 2019.10.24

(71) 申请人 埃米尔·巴登霍斯特

地址 南非帕雷斯

(72) 发明人 埃米尔·巴登霍斯特

(74) 专利代理机构 北京同立钧成知识产权代理有限公司 11205

代理人 杨文娟 臧建明

(51) Int.Cl.

G06F 15/80 (2006.01)

G06F 15/78 (2006.01)

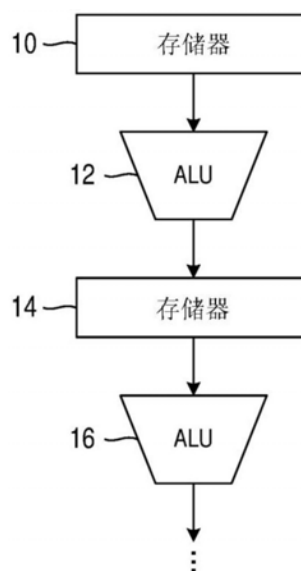
权利要求书2页 说明书5页 附图3页

(54) 发明名称

处理器和操作处理器的方法

(57) 摘要

本公开提供一种包括至少一个核的处理器。所述核包括至少一个输入缓冲器；逻辑单元，其具有输入部和输出部的，其中所述输入部与所述输入缓冲器通信；以及存储单元，其与所述逻辑单元的输出口通信。所述处理器还包括CU（控制单元），其被配置为指导所述核的操作；和通信总线，其被配置为使所述核和所述CU互连。所述CU被配置为通过以下指导所述核的操作：向所述核提供指令，其中所述指令被加载至所述逻辑单元中；将在所述核之一的所述存储单元中存储的值写入所述输入缓冲器。所述CU还被配置为通过经由所述逻辑单元至少部分基于输入缓冲器中的值来提供所述指令的输出，和将所述指令的输出写入所述存储单元，来指导所述核的操作。



1. 一种处理器,包括:
至少一个核,其包括:
至少一个输入缓冲器;
逻辑单元,其具有输入部和输出部,其中所述输入部与所述输入缓冲器通信;以及
存储单元,其与所述逻辑单元的所述输出部通信;
CU(控制单元),其被配置为指导所述核的操作;和
通信总线,其被配置为使所述核和所述CU互连,其中所述CU被配置为通过以下指导所述核的操作:

向所述核提供指令,其中所述指令被加载至所述逻辑单元中;
将在所述核之一的所述存储单元中存储的值写入所述输入缓冲器;
由所述逻辑单元至少部分基于所述输入缓冲器中的值来提供所述指令的输出;和
将所述指令的输出写入所述存储单元。

2. 根据权利要求1所述的处理器,其中所述逻辑单元是ALU(算术逻辑单元)。

3. 根据权利要求1所述的处理器,其中存在多个核。

4. 根据权利要求3所述的处理器,其中所述CU被配置为并行且迭代地操作每个核。

5. 根据权利要求4所述的处理器,其中所述CU被配置为操作所述核,使得只有在所有所述输入缓冲器已经写有所述存储单元中存储的值的条件下,所述存储单元才写有来自所述逻辑单元的输出。

6. 根据权利要求3所述的处理器,其中每个核包括两个输入缓冲器,即第一输入缓冲器和第二输入缓冲器。

7. 根据权利要求6所述的处理器,其中所述逻辑单元被配置为基于所述第一缓冲器和所述第二缓冲器中存储的值来提供输出。

8. 根据权利要求6所述的处理器,其中读入所述输入缓冲器的值来自与所述输入缓冲器相同核的存储单元,或者来自与所述输入缓冲器的核不同的核。

9. 根据权利要求1所述的处理器,其中所述指令包括操作码,或者为操作码的形式。

10. 一种操作具有至少一个核的处理器的方法,所述方法包括:
通过使CU(控制单元)和所述核互连的通信总线,从所述CU向所述核的逻辑单元提供指令;

将所述核的存储单元中存储的值写入连接至所述逻辑单元的输入部的输入缓冲器,所述存储单元连接至所述逻辑单元的输出部;

由所述逻辑单元至少部分地基于所述输入缓冲器中的值来提供所述指令的输出;和
将所述指令的输出写入所述存储单元。

11. 根据权利要求10所述的方法,其中,所述处理器具有至少两个核,并且所述逻辑单元是算术逻辑单元(ALU),所述方法包括:

通过使所述CU和所述核互连的通信总线,从所述CU向每个核的ALU提供指令;

将存储在所述核之一的所述存储单元中的值写入连接至所述ALU的输入部的所述输入缓冲器,所述存储单元连接至所述ALU的输出部;

由所述ALU至少部分基于所述输入缓冲器中的值来提供所述指令的输出;和
将所述指令的输出写入所述存储单元。

12. 根据权利要求11所述的方法, 其中包括将值写入两个输入缓冲器, 即第一输入缓冲器和第二输入缓冲器。

13. 根据权利要求12所述的方法, 其中包括从与所述输入缓冲器相同核的存储单元, 和/或从与所述输入缓冲器的核不同的核写入值。

14. 根据权利要求13所述的方法, 其包括并行且迭代地操作多个所述核。

15. 一种计算机可读介质, 当由具有至少一个核的计算机处理器执行时, 所述计算机可读介质使所述计算机处理器:

通过使CU (控制单元) 和所述核互连的通信总线, 从所述CU向所述核的逻辑单元提供指令,

将所述核的存储单元中存储的值写入连接至所述逻辑单元的输入部的输入缓冲器, 所述存储单元连接至所述逻辑单元的输出部;

由所述逻辑单元至少部分地基于所述输入缓冲器中的值来提供所述指令的输出; 和将所述指令的输出写入所述存储单元。

处理器和操作处理器的方法

技术领域

[0001] 本公开涉及一种处理器和一种操作处理器的方法,尤其涉及一种可以消除或减少竞争条件或存储器锁定的处理器。

背景技术

[0002] 申请人已经观察到常规计算机处理器的指令执行问题,特别是多线程和并行性的问题。由于可以同时执行两条指令,因此它们可以同时特定索引处读取或写入存储器。有问题的原因是程序依赖执行顺序。

[0003] 例如,如果Web服务器托管拥有大量流量的网站,它可能使用多个线程为网页服务,则可能发生以下情况:

[0004] 1. 当Web服务器收到请求时,它读取存储器中包含浏览该网页的总时间的值。

[0005] 2. 然后,Web服务器递增该值,然后更新存储器中的值。

[0006] 更具体地说,线程X和线程Y可能尝试访问值Z。如果值Z为10,并且线程X将值增加5,并且同时线程Y将值增加2,关于线程X还是线程Y是最后写入值Z的线程,将引起竞争。因此,最终该值可能为15或12,这两个值都不正确。最终值应为17。这种现象称为“竞争条件”。解决此问题的主要方法是“锁定”,其中线程X锁定值Z,从而防止线程Y操纵值Z。一旦线程X完成其操作,线程Y便锁定值Z,从而防止线程X访问值Z,从而确保并发性。

[0007] 申请人认为,“锁定”方法的基本问题可能包括:

[0008] 1. 当前未访问特定存储器索引的所有其他线程均被搁置,必须等待释放锁。这导致不计算任何计算的死区时间。

[0009] 2. 这造成了根本上的局限性,其中编排和管理线程并锁定所需的时间和处理能力超过了从大规模多线程获得的优势。

[0010] 3. 竞争条件对于查找和调试非常具有挑战性,并且经常需要对计算机代码进行大的体系结构改变,导致了不必要的复杂性,这导致无法维护。程序中包含的组件越多,存在的bug余地就越大。结果是程序员的工作更加困难。

[0011] 为了解决这个问题,有必要了解数据是如何以空间和时间表示。当前,计算机以可变方式存储数据,这意味着存储器的大部分区域都是可读写的。由于已经存在的所有数据或信息都存在于空间和时间帧中,因此信息的基本性质是不变的,这意味着信息在当前时间帧内不会改变。信息状态只能在正向(即时间上向前)传播。换句话说,对于给定的帧定义,时间上的每个帧都是不可变的。当创建一个新的时间帧时,每个状态都会随着状态的变化而移动到下一帧。

[0012] 因此,过去不能改变,只有未来可以改变。已经存在的或将要存在的所有数据,无论信息是存储在计算机上,还是由人类记忆,或以任何其他方式存储,都必须以某种形式存在于物理世界中。所有物质均受时间规律的约束。因此,可以安全地假设任何构造、思想或信息在物理上都是存在的,这又意味着可以根据时间的性质以不变的方式表示所述信息。

[0013] 申请人希望将该理论应用于处理器体系结构,并且因此应用于包括处理器的任何

计算或电子设备。因为所有信息都可不变地建模,所以可以创建遵循相同的基本时间模型的计算机处理器。因此,竞争条件在理论上不可能以与违反因果关系相同的方式存在。

发明内容

[0014] 因此,本公开提供了一种处理器,包括:

[0015] 至少一个核,其包括:

[0016] 至少一个输入缓冲器;

[0017] 逻辑单元,其具有输入部和输出部,其中所述输入部与所述输入缓冲器通信;以及

[0018] 存储单元,其与所述逻辑单元的所述输出部通信;

[0019] CU(控制单元),其被配置为指导所述核的操作;和

[0020] 通信总线,其被配置为使所述核和所述CU互连,其中所述CU被配置为通过以下指导所述核的操作:

[0021] 向所述核提供指令,其中所述指令被加载至所述逻辑单元中;

[0022] 将在所述核之一的所述存储单元中存储的值写入所述输入缓冲器;

[0023] 由所述逻辑单元至少部分基于所述输入缓冲器中的值来提供所述指令的输出;和

[0024] 将所述指令的输出写入所述存储单元。

[0025] 逻辑单元可以是ALU(算术逻辑单元)和FPU(浮点单元),自定义或集成逻辑电路等。这种类型的处理器体系结构非常适合使用ALU,并且,将参考以ALU形式的逻辑单元进一步描述本公开,但是本公开的范围不必限于使用ALU。

[0026] 可以是多个核。尽管该处理器体系结构仅可在单个核中工作,但通过使用因通过处理器可能提供的并行性而导致的多个核,可以获得更多优势。参考多个核进一步描述本公开,但是本公开可以覆盖单个核或多个核。

[0027] CU可以被配置为并行地操作每个核。CU可以被配置为迭代地操作每个核。CU可以被配置为操作核,使得只有在所有所述输入缓冲器已经写有所述存储单元中存储的值,存储单元才可以写有来自ALU的输出。

[0028] 每个核可以包括两个输入缓冲器,即第一输入缓冲器和第二输入缓冲器。ALU可以被配置为基于所述第一缓冲器和所述第二缓冲器中存储的值来提供输出。

[0029] 读入所述输入缓冲器的值可以来自与所述输入缓冲器相同核的存储单元,也可以来自与所述输入缓冲器的核不同的核。

[0030] 可以有非常多的核。实际的限制是每个核可能需要唯一索引或地址。例如,32位处理器可以提供 $2^{32}-1$ 个核,和64位处理器可以提供 $2^{64}-1$ 个核。实际最大可能为 $2^{\text{位数}}-1$ 个核。所有核可同时使用,无需竞争条件或线程锁定。

[0031] 该指令可以包括操作码或可以是操作码的形式。

[0032] 申请人提出,这可以被概念化为多个时间帧,其中处理周期代表一个时间帧。

[0033] 本公开扩展到一种操作具有至少两个核的处理器的方法,所述方法包括:

[0034] 通过使CU(控制单元)和所述核互连的通信总线,从所述CU向每个核的逻辑单元提供指令;

[0035] 将在所述核之一的存储单元中存储的值写入连接至逻辑单元的输入部的输入缓冲器,所述存储单元连接至所述逻辑单元的输出部;

- [0036] 由所述逻辑单元至少部分基于所述输入缓冲器中的值来提供指令的输出;和
- [0037] 将指令的输出写入所述存储单元。
- [0038] 所述方法可以包括以ALU(算术逻辑单元)的形式操作逻辑单元。
- [0039] 所述方法可以包括将值写入两个输入缓冲器,即第一输入缓冲器和第二输入缓冲器。可以从与所述输入缓冲器相同核的存储单元和/或从与所述输入缓冲器的核不同的核写入值。
- [0040] 所述方法可以包括并行地操作多个核。
- [0041] 所述方法可以由上面定义的处理器来实现。
- [0042] 本公开扩展到一种计算机可读介质,当其被计算机处理器执行时,根据所述方法操作处理器。
- [0043] 所述处理器和方法背后的理论和模型可以如下(参考示例值和数据)。
- [0044] 表1代表存储器状态,表2代表指令。

[0045]

#	C0	C1	C2	C3	C4	C5	C6
1	10	10					
2	20						
3							

[0046] 表1

[0047]

#	C0	C1	C2	C3	C4	C5	C6
1	L10	L10					
2	C0+C1						
3							

[0048] 表2

[0049] 在表1-2中,每一列(C0,C1...)代表存储器地址。数据类型与模型无关,但是在此示例中,使用了无符号整数。每行代表计算周期,其被称为帧。从理论上讲,帧和存储器的数量可以是无限的。首先,模型在帧之间移动。每帧只能写入一次。帧只能创建而不可以更改,因此它是不可变的。这紧密地模仿了时间的本质。

[0050] 帧1:引用指令帧1,C0包含将字面量10加载到C0的存储器索引中的指令。C1同样适用。因此,存储器的状态为C0和C1中的字面量10。

[0051] 帧2:因为帧是不可变的,所以前一帧只能读取而不能写入。牢记这一点,我们可以看到帧2中的唯一指令是单元格C0。该单元格中的指令是加法运算符,该运算符取消引用地址C0和C1以将两个值加在一起,在当前帧中将结果写入C0。

[0052] 因此,该模型计算出的算法为 $10+10$,结果为20。由于每个帧都是不可变的(仅在一个方向上传播),因此可以同时计算对于当前帧的每个存储单元的所有计算,而不会违反并发性或创造性竞争条件。

[0053] 对于每个帧具有专用存储单元和ALU将是不切实际且不可扩展的,这就是上面定义的处理器和方法迭代地使用相同的存储单元和ALU的原因。

附图说明

[0054] 现在将参考附图通过示例进一步描述本公开。

- [0055] 在附图中：
- [0056] 图1示出了概念处理器模型；
- [0057] 图2示出了图1的模型的环形版本；
- [0058] 图3示出了根据本公开的处理器的示意图；和
- [0059] 图4示出了根据本公开的操作处理器的方法的流程图。

具体实施方式

[0060] 提供本公开的以下描述作为本公开的使能教导。相关领域的技术人员将认识到，可以对所描述的实施例做出许多改变，同时仍然获得本公开内容的有益结果。还将显而易见的是，通过选择本公开的一些特征而无需利用其他特征，可以实现本公开的一些期望益处。因此，本领域技术人员将认识到，对本公开的修改和适应是可能的，并且在某些情况下甚至是期望的，并且是本公开的一部分。因此，提供以下描述作为本公开的原理的说明，且不是对其的限制。

[0061] 本公开提供了处理器100 (在图3中示出) 和方法200 (在图4中示出)。然而，图1-2虽然不是本公开的必要部分，但是提供了对处理器100和方法200实现的模型的理解，并且与公开发明内容中的描述相联系。

[0062] 图1示出了概念上的处理器。通过将每个存储位置10、14馈送至ALU 12、16中，每个存储位置10、14中的所有值同时将结果馈送到新的存储帧10、14中；然而，不可能有无限数量的存储帧，也不可能有无限数量的ALU 12、16，这使得该设计不切实际。

[0063] 为了实际地实现这一点，如图2所示，可以通过将ALU输出馈送到存储器18中，然后将存储器输出馈送到(再次馈送到ALU 20中的)缓冲器中来模拟期望的行为，导致模仿完全相同行为的循环。之所以模仿该行为，是因为与当前帧中的值相关的唯一帧是前一帧。因此，紧接在前一帧之前的所有前一帧与执行状态无关，因此可以丢弃。

[0064] 图3示出了根据本公开的处理器100。处理器100包括多个核102 (仅示出了其中三个：核1，核2和核n)。然而，应当明确，可能存在许多核，甚至比通常具有2-16个核的当前可用的消费级处理器高几个数量级。

[0065] 每个核102包括两个缓冲器104，即第一缓冲器104.1和第二缓冲器104.2。可能存在于其他实现，其中核102仅包括单个缓冲器或两个以上的缓冲器，但是鉴于操作码和逻辑经常仅应用于两个输入，因此对于该示例，具有两个输入缓冲器104是适当的。处理器100具有至少一个CU 101，其被配置为按以下所述地指导核102的操作。

[0066] 每个核102包括ALU 106。ALU 106本身可以在某种程度上是常规的，因为它具有连接至缓冲器104的输入部或一对输入部，和连接至存储单元108的输出部。基于来自输入缓冲器104的输入，ALU 106被配置为实现通常以操作码形式出现的指令以产生被写入存储单元108的输出。

[0067] 通信总线110或通信总线的网络与CU 101和核102以及根据需要任何其他外围组件互连。

[0068] 参照图4所示的操作处理器的方法200进一步描述处理器100。然而，应当理解，处理器100可以被配置为实现不同的方法，并且方法200可以在不同配置的处理器上实现。

[0069] 方法200示出了单个循环，但其可以是无限环形的。一个循环的实际示例如下：

[0070] 框202:CU 101向每个核102提供指令。该指令被提供作为ALU术语中理解的操作码。可以向不同的ALU 106提供不同的指令。

[0071] 框204:将ALU 102加载到每个ALU 106中作为操作码。

[0072] 框206:读取(如当前存储在存储单元108中的)前一帧的所需值。

[0073] 框208:从存储单元读取的所需值通过通信总线110被写入缓冲器104。

[0074] 框210:ALU 106基于从缓冲器104的输入和所提供的特定操作码来计算输出。

[0075] 框212:将输出写入关联的存储单元108。

[0076] 该方法200无限循环(在框214处)。该无限循环提供来自(在存储单元108中的)当前周期的输出的存储以及来自缓冲器104中的先前周期的所有或一些输出的存储。根据该模型,所需的就当前周期和上一个周期的输出。然后在下一周期中再循环缓冲器104、ALU 106和存储单元108。

[0077] 申请人认为,示例性的处理器100和方法200具有几个优点:

[0078] 存储器本身(以存储单元108和/或缓冲器104的形式)与核102组合,从而大幅度减少了将值从存储器移入核所需的时间,因为存储器和核可以在同一芯片中。因此,无需从外部存储源检索数据。

[0079] 处理器100可以同时执行大规模并行的多个线程。

[0080] 多个线程可以使用相同的特定存储器位置同时执行,而不会发生互斥或锁定或竞争条件。

[0081] 存储器可以只在一个周期写入一次,此后它将变为“只读”。处理器状态始终沿一个方向移动。

[0082] 每个存储位置都可以同时被处理。

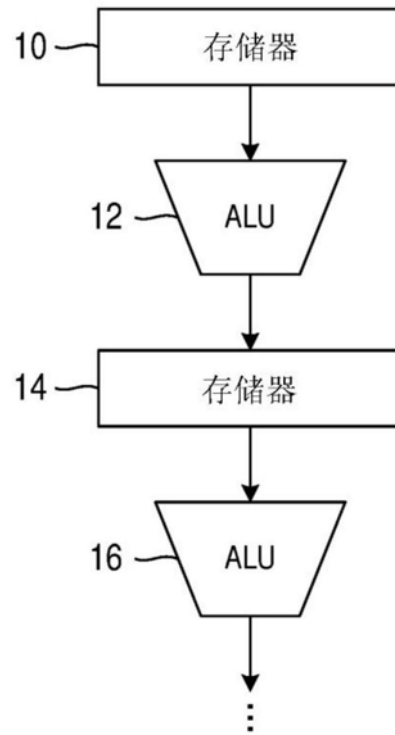


图1

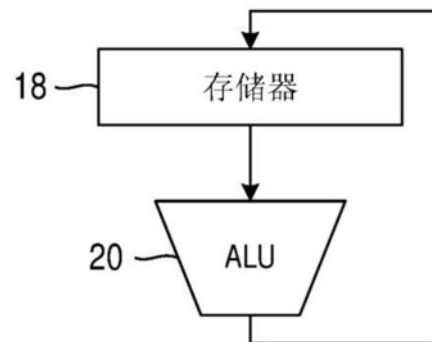


图2

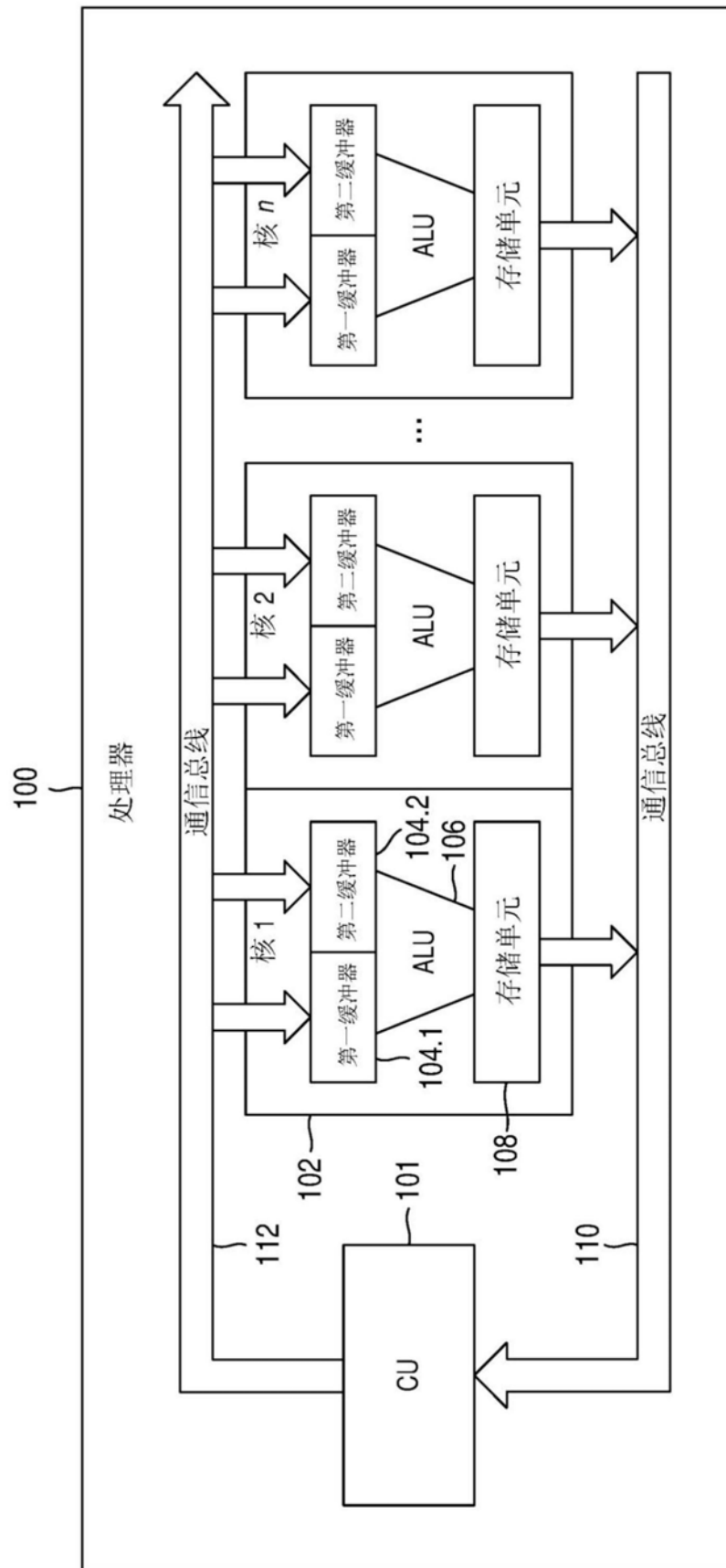


图3

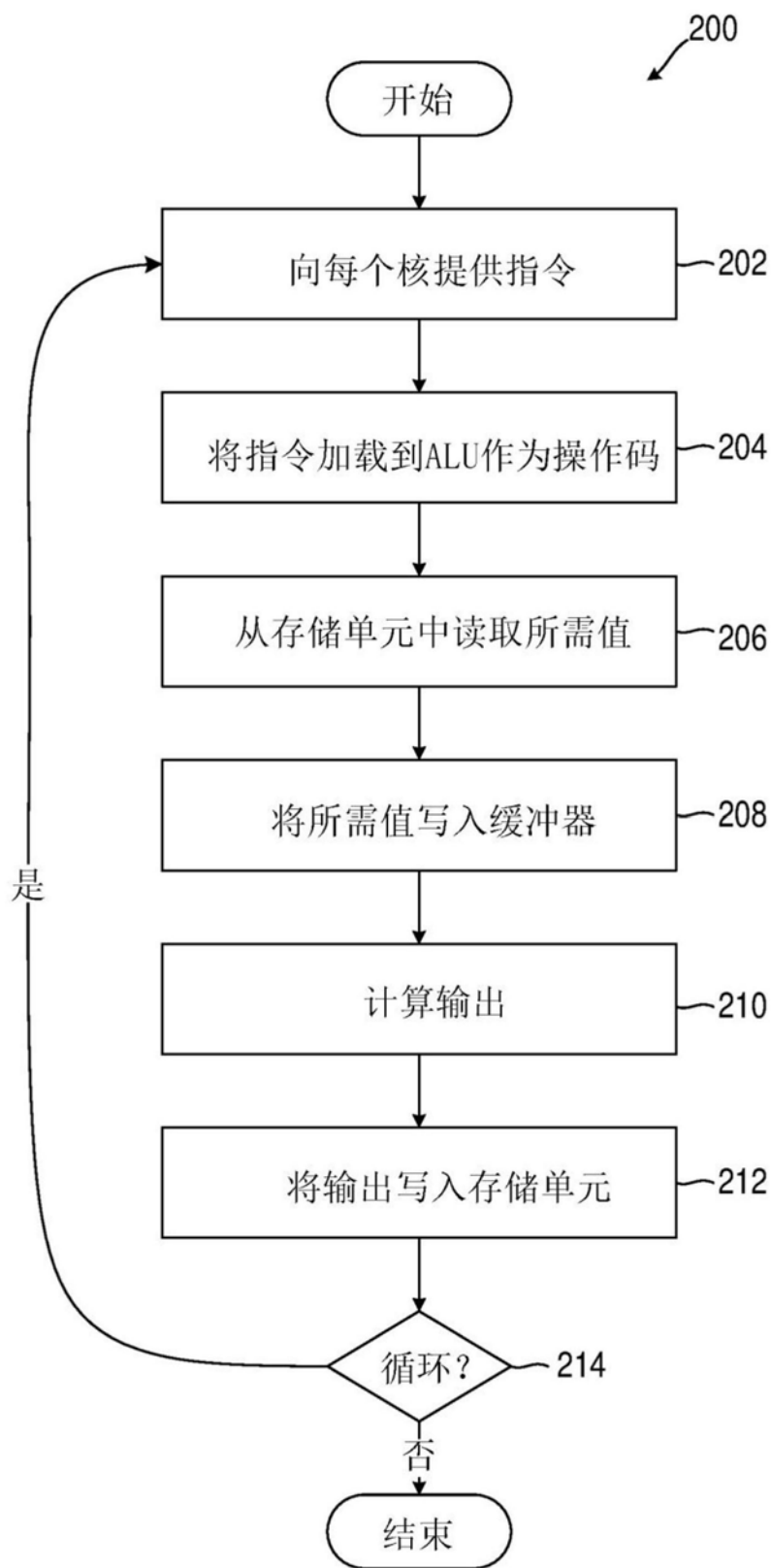


图4